Textbooks: Under Inspection

Michael de Raadt, Richard Watson
Department of Mathematics and Computing
University of Southern Queensland
Toowoomba, Queensland, Australia
{deraadt,rwatson}@usq.edu.au

Mark Toleman
Department of Information Systems
University of Southern Queensland
Toowoomba, Queensland, Australia
markt@usq.edu.au

Abstract

Textbooks are an important resource in tertiary education, particularly in Introductory Programming. In many university courses they form the basis of curriculum. But how good are the texts being used? Can they be relied on for skills required for students' later study and employment? Do they earn the high price tag publishers place on them?

The 2003 Introductory Programming Census of Australian and New Zealand universities' courses revealed texts in use. This paper objectively compares these texts, differentiating them according to content and features, ACM/IEEE curriculum compliance and problem solving approach. This review found large variation in the texts reviewed. Many texts rely on examples to fill content. Few texts attempt to comply with curricula recommendations. Few contain a continuous instruction in problem solving.

Keywords: introductory programming, textbooks

1 Introduction

The most recent Census of eighty-five introductory programming courses in Australian and New Zealand universities (de Raadt, Watson and Toleman 2004), discovered that forty-nine texts were being prescribed to students. This paper attempts to compare these texts.

This comparison is motivated by the following factors.

- Many instructors who participated in the Census reported that they were constantly engaged in teaching and related activities. It is likely that such instructors would lack the time to undertake a thorough comparison of a complete set of available texts. The authors believe it would be valuable to create a list of such texts as a resource for instructors and the public, which is without publisher bias.
- There is a lack of any existing equivalent study which focuses on current texts available in Australia and New Zealand. Some research reflects text impact on a migration between languages or teaching approaches (Doube 2000; Becker 2001). Other papers refer to the role and structure of texts used in teaching (Lister 2000; Kuittinen and Sajaniemi 2004; Simon 2004). Means (1988) compares a number of texts and reports a lack of change in texts over time.
- The ACM/IEEE 2001 curricular guidelines (ACM IEEE Joint Task Force on Computing Curricula 2001) provide a strong recommendation for course content. It is desirable to discover if any texts explicitly follow these guidelines, or if any texts can otherwise be used with the guidelines.
- It is desirable to discover the problem solving approaches and depth present in current texts.

2 Methodology

A pilot study was conducted based on twelve representative texts. The study was used to identify the important features of texts that could be used to make meaningful comparisons and to ensure the reviewers of the texts were consistent in their appraisal of these features. The features are described below.

The content of each text was divided into chapter content, language reference, glossary, bibliography and index. The size of these elements was measured in complete pages. Within each chapter, the proportion occupied by exercises and examples can be separated from other chapter content. For the purposes of comparison, an example is seen as a complete, continuous item of source code. Exercises contained within a chapter and at the end of a chapter were measured together. Exercises range from reflective questions, which may consist of a single line of text, through to

projects which may occupy several pages. Exercises and examples were measured in partial pages. Frontmatter and preface material was not measured, although instructor information is usually contained in the preface and this was compared.

The ACM/IEEE 2001 curricular guidelines suggest several tracks that involve introductory programming courses listed as follows.

- CS101I. Programming Fundamentals (Imperative-first Track)
- CS111I. Introduction to Programming (Imperative-first Track)
- CS101O. Introduction to Object-Oriented Programming (Objects-first Track)
- CS111O. Object-Oriented Programming (Objects-first Track)
- CS111F. Introduction to Functional Programming (Functional-first Track)
- CS101B. Introduction to Computer Science (Breadth-first Track)
- CS111A. Introduction to Algorithms and Applications (Algorithms-first Track)
- CS111H. Introduction to the Computer (Hardware-first Track)

Within these courses, core units are recommended which cover introductory aspects relevant to the containing course. The extent to which each text covered these core units was recorded. These units are listed as follows.

- Discrete Structures (DS1-6)
- Programming Fundamentals (PF1-4)
- Algorithms and Complexity (AL1-3,5-6)
- Architecture and Organisation (AR1-4)
- Operating Systems (OS1,3)
- Net-Centric Computing (NC1)
- Programming Languages (PL1,3-8)
- Human Computer Interaction (HC1)
- Graphics and Visual Computing (GV1)
- Social and Professional Issues (SP1-2,4-6)
- Software Engineering (SE1-3,5-6)

Texts vary significantly in the way content was presented. The presence of various layout aspects was determined to allow instructors to choose texts that match their needs and preferences. Layout aspects measured include location of exercises (within chapters, at end of chapters, or both), the presence of more than one scale of exercise (review questions, questions for paper exercises and/or coding exercises), colour coding and visual separation of sections within chapters.

The presence of extra teaching materials (not directly related to programming language syntax or problem solving) was measured. This included practice hints, common error warnings, biographies of historical persons and other historical information, interviews with current industry related persons, real world industry based case studies and other non-programming facts.

Resources available for use by instructors or students were noted. This included a language reference, code examples and/or a compiler. The location of such resources was also noted as being within the text, on a CD that accompanied the text, on a website, or from some other source (for example, by email from the author or publisher). The availability to students and instructors of solutions to exercises was gathered. Availability of other instructor resources including a laboratory manual, instructor's notes, lecture slides and a test question bank was determined.

The approach and depth of problem solving content was noted. Other general notable aspects were recorded.

The forty texts reviewed by the authors contained over 31,000 pages, most of which have been individually inspected, sometimes more than once (a page turning marathon!) While every attempt has been made to maintain accuracy, the amount of visual information and range of presentation styles virtually guarantees that errors of omission will have occurred. Counting part pages of examples and exercises that were distributed throughout chapters is also problematic. This was performed with a combination of visual estimation and measurement; the error bound for these features is at best ±10%.

3 Results and Discussion

Forty-nine texts were reported by the 2003 Census. Some instructors reported that they did not prescribe a text and relied on their own materials. Some courses used more than one text. All reported texts were included. Forty texts were made available by local text distributors. All texts prescribed during 2003 are listed in Appendix 1 including language (where applicable), count of courses where the text is prescribed, the number of students undertaking those courses and if the text was made available to this study. Each text was given a unique number which can be used as a key for matching between later tables. The overall picture of an average text is shown in Table 1.

Average pages	775.4
Average chapters	16.3
Average examples	113
Average example length	0.92 pages
Average exercises	470.4
Average exercise length	0.17 pages

Table 1: Overall text averages.

The most widely used text is Lesley Anne Robertson's "Simple Program Design". This is not a text targeted at a specific language, but rather covers problem solving aspects related to programming. This text is used together with a language based text in most cases.

When choosing a text an instructor may appreciate more code examples. The instructor may want exercises which can be given to students for practical work. They may want a text which includes a language reference. The inclusion of a compiler with the text may be desirable. Appendix 3 shows the proportions of content separated into examples, exercises, other chapter content, language reference, glossary, bibliography, other appendix content and index. Appendix 4 shows the features of each text including the number of exercises and examples, the presence or absence of various layout features, the inclusion and location of a language reference, code examples and compiler, the availability of solutions to students and instructors, and the availability of various instructor resources.

The ACM/IEEE curricular guidelines (ACM IEEE Joint Task Force on Computing Curricula 2001) suggest eight introductory programming courses with a number of core units specified in each. Appendix 5 shows how each text could be applied to these courses. Where the description of a unit matched the content of a topic present in a text, that unit was marked as present for the text. The proportion of units covered is shown as a fraction of the total course core units. Different texts approached these topics from different perspectives and to differing depths. These differences were not measured.

Problem solving instruction was presented in varying degrees between texts. Some texts avoid problem solving as a specific topic altogether. Some authors rely on large numbers of examples and contain little explicit instruction of problem solving. One author bluntly stated "Students learn to program by example" (Gerard Sparke "The Java Way"). Some texts offer a brief mention of algorithmic problem solving in an early chapter, but this teaching is not obviously integrated in the remainder of the text. Other texts offer high level systems analysis or software engineering but little algorithmic problem solving; object-oriented software engineering is a common topic. There were a small number of texts which raise problem solving and attempt to integrate this teaching throughout the text using case studies and examinations of problems. Of note were:

- D.S. Malik "C++ Programming: From Problem Analysis to Program Design";
- Jeri R. Hanly & Elliot B. Koffman "C Program Design for Engineers";
- Simon Thompson "Haskell: The Craft of Functional Programming";
- Elliot B. Koffman & Ursula Wolz "Problem Solving with Java";
- C. Thomas Wu "Introduction to Object-Oriented Programming with Java"; and
- Walter J. Savitch "Problem Solving with C++: The Object of Programming".

Different types of texts are being used according to the 2003 census. Most texts were academic in their approach including instruction, examples and exercises. At least one reported text could be considered a commercial text

consisting of language syntax instruction for a member of the general public but was notably lacking exercises. One reported text could be considered as a reference only with an extensive language summary and no instruction or exercises. In both cases where this text is used it is in conjunction with another text.

4 Conclusions and Recommendations

It was not obvious in most texts that authors had attempted to follow a standard curricular model presented by the ACM, IEEE or other organisation. The ACM/IEEE 2003 curriculum specifies a number of courses and the content of these courses has been matched to texts within this study. An author attempting to cover many courses may wish to consider including instruction on certain core units common to several of the courses as shown in Table 2.

Units	Courses using units
PF1, PF2, PF2, SP1	8 (all)
AL3, AL5 PL4	7
PL1, PL5, SE1	6
PF4, SE3	5
AR2, PL6	4

Table 2: Units common to four or more courses in the ACM/IEEE curriculum guidelines.

Many texts rate poorly in the matching to the ACM/IEEE curriculum guidelines. This may be because these text's authors focus on presenting a programming language in as much detail as possible rather than applying a more holistic approach to programming. One of the tracks of the ACM/IEEE curriculum guidelines is suggested as Breadth-first, yet even the introductory programming courses present in other tracks appear to cover a broader range of topics than those present in most texts. For this reason texts with a more general computer science focus rated higher. Some of these texts were independent of any programming language and some included a shallow teaching of one particular language.

Who writes texts and why? Most authors represented in this study are writing texts as a source of income. Some of the texts are in fourth and fifth editions indicating not only the success of the texts, but also the willingness of the authors to keep them up to date. Texts of this type are probably targeted at a wide audience and may attempt to cover a broader range of topics. Some texts are constructed by instructors for use with their own students. These texts are likely to be focused on the author's current curriculum, but may be adopted by other instructors if the content is considered appropriate. It may be useful for instructors considering a text to determine which of these two categories it falls into; publishers should be able to answer this question.

What makes a good textbook? This question does not have a simple or a single answer. To some extent it depends upon how the book it to be used and who is the audience. The following is a set of features whose inclusion we consider to be very important, if not mandatory.

Effective use of graphical devices

While analysing the content of several thousand pages, it became apparent that texts which visually separated exercises and examples from other chapter content, using colour and other graphical means, were much easier to read. This aspect may not reflect the pedagogical significance of the texts, but it was quite clear that several texts poorly distinguish sections using formatting. A textbook is a collection of many different kinds of information, as described in Appendix 3. Unlike a novel, a textbook is not read linearly but rather consumed in small chunks with much searching backward and forward for information. A good text facilitates this navigation by appropriate differentiation of the various types of material. While it is clearly possible to overdo the use of colour in particular, appropriate use of such techniques as background colours, boxing of information, marginal notes, sidebars, and icons can significantly enhance the usability of a text. It can also make a book a pleasure to read.

Well chosen and presented examples

Most reviewed texts provided some examples of code. Some were presented in complete continuous blocks while others presented only fragments. Examples should concisely illustrate a technique. They should include line numbers for reference, though should preferably be as self-contained as possible, not requiring the reader to keep referring back to the accompanying text discussion. Better examples will often include the author's comments, maybe accompanied with some lines and arrows just like the typical classroom blackboard example.

Exercises

A distinction was made between texts with exercises located within chapters, at the end of chapters and texts taking both approaches. Incorporating exercises within chapters encourages readers to reflect on content almost immediately. This may have a positive pedagogical benefit, especially for students undertaking independent or external study. Having exercises at the end of a text allows instructors to suggest practical work for students which can be located easily. This may be beneficial in face-to-face tutorial situations. Having a range of exercises from reflective activities to programming projects may also aid an instructor. If a text were to be constructed to suit different types of students while assisting instructors in their work, all these aspects would need to be combined.

Solutions
Solutions to texts were presented in a number of ways. Some texts presented no solutions, some presented a partial set of solutions and some presented all solutions. Some texts posed restrictions on access to solutions allowing only instructors to access solutions not in the text. Obviously having solutions is more important to students studying independently. The presence of selected solutions to questions and exercises can be a boon to students and instructors alike.

Biographies
These are introductory textbooks. It behoves the authors to include some background information about the discipline; well chosen biographies, and possibly small interviews do this very well and have the added advantage of motivating students (and adding some light relief from a difficult subject). Surprising few texts included this feature; those that did tended to have a broader focus.

Case studies
Explicitly showing how to progress from a requirement specification, however simple, through to a design or algorithm, then on to the final coded program is an excellent way to teach problem solving skills. Students need to see the intermediate steps, explained by an expert, in order to build their own competence. Adding one case study per chapter is a good aim, though few texts have adopted this approach.

This study has presented a relatively brief examination of approaches to problem solving instruction. However this is a key pedagogical topic, and much remains to be done in analysing and comparing the approaches taken by various authors. Future work is planned to address this issue.

5 References

ACM IEEE Joint Task Force on Computing Curricula (2001). Computing Curricula 2001, Computer Science. Available online: http://www.computer.org/education/cc2001/final/index.htm, Accessed: 17th August, 2004.

Becker, B. W. (2001). "Teaching CS1 with Karel the Robot in Java." ACM SIGCSE Bulletin **33**(1): 50-54.

de Raadt, M., R. Watson and M. Toleman (2004). Introductory Programming: What's happening today and will there be any students to teach tomorrow? Proceedings of the Sixth Australasian Computing Education Conference (ACE2004), Dunedin, New Zealand, Australian Computer Society.

Doube, W. (2000). The impact on student performance of a change of language in successive introductory computer programming subjects. Proceedings of the on Australasian computing education conference, Melbourne Australia, ACM Press, New York, NY, USA.

Kuittinen, M. and J. Sajaniemi (2004). Teaching roles of variables in elementary programming courses. Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education, Leeds, United Kingdom, ACM Press.

Lister, R. (2000). On Blooming First Year Programming and its Blooming Assessment. Proceedings of the on Australasian computing education conference, Melbourne Australia, ACM Press, New York, NY, USA.

Means, H. W. (1988). A content analysis of ten introduction to programming textbooks. Proceedings of the nineteenth SIGCSE technical symposium on Computer science education, Atlanta, Georgia, United States, ACM Press.

Simon (2004). The cryptic crossword puzzle as a useful analogue in teaching programming. Proceedings of the sixth conference on Australian computing education, Dunedin, New Zealand, Australian Computer Society, Inc., Darlinghurst, Australia, Australia.

6 Appendices

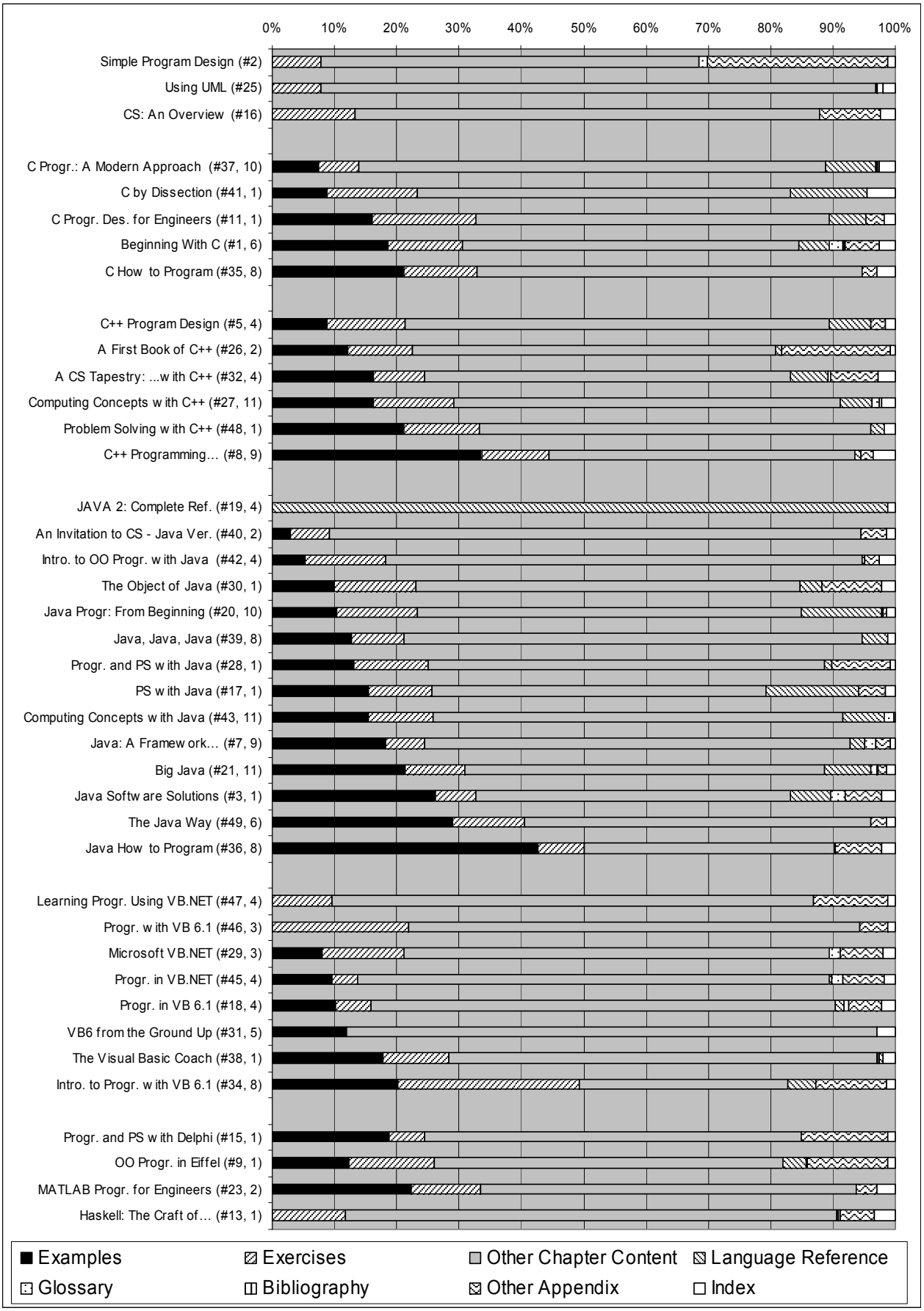
Appendix 1. List of Books and Publication Details

#	Title	Author	Publ.	Language	Latest Ed	Year	ISBN	Courses	Students	Avail.
2	Simple Program Design	Lesley Anne Robertson	9	None	4th	2004	0-17-010704-3	12	2294	Yes
3	Java Software Solutions	Lewis & Loftus	1	Java	3rd	2003	0-201-78129-8	8	1470	Yes
7	Java: A Framework for Programming and Problem Solving	Kenneth A. Lambert & Martin Osborne	9	Java	2nd	2002	0-53-438277-0	5	1291	Yes
4	Introduction to Programming Using VB.NET	David I. Schneider	8	VB.NET	5th	2003	0-13-030657-6	5	770	No
21	Big Java	Cay Horstmann	11	Java	1st	2001	0-471-40248-6	4	1302	Yes
12	Java: An Introduction to Computer Science & Programming	Walter Savitch	8	Java	3rd	2001	0-13-101378-5	3	980	No
11	C Program Design for Engineers	Jeri R. Hanly & Elliot B. Koffman	1	C	2nd	2001	0-201-70871-X	3	229	Yes
19	JAVA 2: The Complete Reference	Herbert Schildt	4	Java	5th	2002	0-07-222420-7	2	1200	Yes
26	A First Book of C++: From Here to There	Gary J. Bronson	2	C++	2nd	2000	0-53-436801-8	2	700	Yes
8	C++ Programming: From Problem Analysis to Program Design	D.S. Malik	9	C++	1st	2002	0-61-906213-4	2	640	Yes
17	Problem Solving with Java	Elliot B. Koffman & Ursula Wolz	1	Java	2nd	2002	0-321-15486-X	2	560	Yes
30	The Object of Java	David D. Riley	1	Java	Blue J Ed	2003	0-321-16854-2	2	430	Yes
13	Haskell: The Craft of Functional Programming	Simon Thompson	1	Haskell	2nd	1999	0-201-34275-8	2	383	Yes
29	Microsoft Visual Basic .NET	Diane Zak	3	VB.NET	Reloaded	2004	0-619-21287-X	2	370	Yes
18	Programming in Visual Basic 6.0	Julia Case Bradley & Anita C. Millspaugh	4	VB6	1st	2002	0-07-251874-X	2	260	Yes
34	Introduction to Programming with Visual Basic 6.0	David I. Schneider	8	VB6	4th	2004	0-13-142707-5	2	145	Yes
35	C How to Program	Harvey M. Deitel & Paul J. Deitel	8	C	4th	2004	0-13-142644-3	2	100	Yes
20	Java Programming: From the Beginning	K. N. King	10	Java	1st	2000	0-393-97437-5	1	700	Yes
32	A Computer Science Tapestry: Exploring Computer Science with C++	Owen Astrachan	4	C++	2nd	2000	0-07-246536-0	1	700	Yes
44	Programming, Problem Solving & Abstraction with C	Alistair Moffat	7	C	1st	2002	1-74-103080-3	1	700	No
6	Structured & Object-Oriented Problem Solving Using C++	Andrew C. Staugaard	8	C++	3rd	2002	0-13-028451-3	1	600	No
33	An Introduction to Computing with Haskell	Manuel Chakravarty and Gabriele Keller	7	Haskell	1st	2003	1-74-009404-2	1	600	No
38	The Visual Basic Coach	Jeff Salvage	1	VB6	1st	2001	0-201-74549-6	1	500	Yes
42	Introduction to Object-Oriented Programming with Java	C. Thomas Wu	4	Java	2nd	2001	0-07-255133-X	1	500	Yes
14	Objects First with Java: A Practical Introduction Using BlueJ	David J. Barnes & Michael Kolling	8	Java	1st	2003	0-13-044929-6	1	350	No
22	Java Genesis	Roger Duke & Eric Salzmann	1	Java	2nd	1999	1-74-091111-3	1	350	No
28	Programming and Problem Solving with Java	James M. Slack	1	Java	1st	2000	0-53-437486-7	1	350	Yes
9	Object-Oriented Programming in Eiffel	Raymond A. Weedon & Peter G. Thomas	1	Eiffel	2nd	1998	0-20-133131-4	1	340	Yes
37	C Programming: A Modern Approach	K. N. King	10	C	1st	1996	0-393-96945-2	1	300	Yes
43	Computing Concepts with Java Essentials	Cay Horstmann	11	Java	3rd	2003	0-471-46900-9	1	280	Yes
24	Introduction to Java Programming with Sun ONE Studio 4	Y. Daniel Liang	8	Java	1st	2003	0-13-009258-4	1	253	No
25	Using UML: Software Engineering with Objects and Components	Rob Pooley & Perdita Stevens	1	None	1st	2000	0-201-64860-1	1	253	Yes
5	C++ Program Design	James P. Cohoon & Jack W. Davidson	4	C++	3rd	2001	0-07-256040-1	1	235	Yes
48	Problem Solving with C++: The Object of Programming	Walter J. Savitch	1	C++	5th	2004	0-321-26865-2	1	230	Yes
40	An Invitation to Computer Science - Java Version	G. Michael Schneider & Judith Gersting	2	Java	1st	2000	0-53-437488-3	1	203	Yes
1	Beginning With C	Ron House	6	C	1st	1994	0-17-008821-9	1	200	Yes
15	Programming and Problem Solving with Delphi	Mitchell C. Kerman	1	Delphi	1st	2002	0-201-70844-2	1	200	Yes
16	Computer Science: An Overview	J. Glenn Brookshear	1	None	7th	2003	0-201-78130-1	1	200	Yes
41	C by Dissection: The Essentials of C Programming	Al Kelley & Ira Pohl	1	C	4th	2000	0-201-71374-8	1	200	Yes
23	MATLAB Programming for Engineers	Stephen J. Chapman	2	Matlab	2nd	2002	0-534-39056-0	1	170	Yes
36	Java How to Program	Harvey M. Deitel & Paul J. Deitel	8	Java	5th	2003	0-13-101621-0	1	130	Yes
45	Programming in Visual Basic .NET	Julia Case Bradley & Anita C. Millspaugh	4	VB.NET	4th	2003	0-07-293870-6	1	120	Yes
47	Learning Programming Using Visual Basic.Net	William E. Burrows, Joseph D. Langford, Johny K. Johansson	4	VB.NET	4th	2003	0-07-293871-4	1	100	Yes
10	C++: An Introduction to Computing	Joel Adams & Larry R. Nyhoff	8	C++	3rd	2002	0-13-091426-6	1	87	No
31	Visual Basic 6 from the Ground Up	Gary Cornell	5	VB6	1st	1998	0-078-82508-3	1	80	Yes
39	Java, Java, Java Object-Oriented Problem Solving	Ralph Morelli	8	Java	2nd	2002	0-13-033370-0	1	60	Yes
27	Computing Concepts with C++ Essentials	Cay Horstmann	11	C++	3rd	2002	0-471-16437-2	1	50	Yes
46	Programming with Visual Basic 6.0	Diane Zak	3	VB6	Enhanced	2001	0-619-06204-5	1	50	Yes
49	The Java Way : An Introduction to Programming in Java	Gerard Sparke	6	Java	1st	2003	1-74-103191-5	1	40	Yes

Appendix 2. List of Publishers and Associated Distributors

#	Publisher	Aus/NZ Distributor
1	Addison-Wesley	Pearson Education
2	Brooks-Cole	Thompson Learning
3	Course Technology	Thompson Learning
4	McGraw-Hill	McGraw-Hill
5	McGraw-Hill Osborne Media	McGraw-Hill
6	Nelson ITP	Thompson Learning
7	Pearson SprintPrint	Pearson Education
8	Prentice Hall	Pearson Education
9	Thomson Nelson	Thompson Learning
10	W.W. Norton Company	Wiley
11	Wiley	Wiley

Appendix 3. Content Proportions



Appendix 4. Features in texts

#	Language	Courses	Pages	Chapters	Examples	Exercises	Exercises Within Chapter	Exercises at end of chapters	Multiple scales of questions	Projects/Assignments	Chapter summaries	Colour coding	Visual layout for section separation	Practice hints	Common error warnings	Historical/Biographies of historical people	Interviews/Biographies from current industry	Case studies (industry based)	Non programming facts	Guide to use of chapters	Language reference	Code Examples	Compiler	Solutions for student	Solutions for instructor	Lab Manual	Lect Slides	Notes	Test Bank
2		Simple Program Design	345	13	0	91		✓			✓		✓								🕮								
16		CS: An Overview	562	12	0	883						✓												all	all		✓	✓	✓
25		Using UML	256	20	0	208	✓	✓			✓									✓					all			✓	
1	C	Beginning With C	570	19	123	370					✓	✓									🕮	⊙		some	some				
11	C	C Progr. Des. for Engineers	673	13	98	527		✓	✓	✓	✓		✓		✓			✓				💻		some	some			✓	✓
35	C	C How to Program	1255	30	264	652		✓	✓		✓	✓	✓	✓	✓							🕮💻	⊙	all	all				
37	C	C Progr.: A Modern Approach	659	26	65	310		✓			✓				✓						🕮	🕮💻							
41	C	C by Dissection	630	15	93	294		✓			✓			✓	✓					✓	🕮	🕮							
44	C	Progr., PS & Abstr. with C	Not available at time of publication																										
5	C++	C++ Program Design	948	15	186	935	✓	✓	✓	✓	✓		✓	✓	✓	✓			✓	✓	🕮	🕮⊙	⊙				✓		
6	C++	Struct. & OO PS Using C++	Not available at time of publication																										
8	C++	C++ Programming…	1104	18	167	357		✓	✓		✓	✓	✓								🕮	🕮💻		some	all		✓	✓	
10	C++	C++: An Intro. to Comp.	Not available at time of publication																										
26	C++	A First Book of C++	798	14	142	445	✓				✓			✓	✓						🕮	🕮💻		some	all	✓	✓		✓
27	C++	Computing Concepts with C++	765	18	110	593		✓	✓		✓		✓	✓	✓	✓			✓	✓	🕮	🕮💻		some	all	✓	✓		
32	C++	A CS Tapestry: ...with C++	845	13	155	143	✓	✓	✓		✓	✓	✓	✓		✓				✓	🕮⊙	⊙💻			all				
48	C++	Problem Solving with C++	937	18	145	580	✓	✓	✓	✓	✓	✓	✓	✓	✓					✓		💻		all	all		✓	✓	✓
3	Java	Java Software Solutions	793	12	171	509		✓	✓		✓	✓	✓							✓		🕮💻	⊙		all		✓	✓	✓
7	Java	Java: A Framework…	683	18	92	486	✓		✓	✓	✓	✓	✓	✓		✓				✓	🕮		⊙	some	some			✓	
12	Java	Java: An Intro to CS & Progr.	Not available at time of publication																										
14	Java	Objects First with Java	Not available at time of publication																										
17	Java	PS with Java	856	10	145	636	✓	✓	✓		✓		✓		✓						🕮	🕮	⊙	some	some				
19	Java	JAVA 2: Complete Ref.	1154	0	0	0															🕮	🕮							
20	Java	Java Progr: From Beginning	777	14	66	715	✓	✓	✓	✓	✓		✓		✓					✓	🕮	🕮⊙	⊙	some	some		✓	✓	✓
21	Java	Big Java	1197	25	210	779		✓	✓		✓		✓	✓	✓				✓	✓		💻		some	all	✓	✓		✓
22	Java	Java Genesis	Not available at time of publication																										
24	Java	Intro. to Java Progr. with Sun…	Not available at time of publication																										
28	Java	Progr. and PS with Java	1137	14	180	838	✓	✓			✓	✓	✓	✓								🕮💻							✓
30	Java	The Object of Java	717	15	110	168		✓	✓			✓	✓	✓						✓	🕮	⊙💻	⊙		all		✓		
36	Java	Java How to Program	1447	25	298	636		✓	✓	✓	✓	✓	✓	✓	✓			✓		✓				some	all	✓	✓		
39	Java	Java, Java, Java	866	17	215	651	✓	✓			✓		✓	✓	✓	✓				✓	🕮	🕮	⊙	some	some		✓		
40	Java	An Invitation to CS - Java Ver.	734	14	38	338			✓		✓	✓	✓		✓				✓							✓			
42	Java	Intro. to OO Progr. with Java	863	17	47	450	✓	✓				✓	✓			✓				✓		🕮⊙	⊙	some	all		✓		✓
43	Java	Computing Concepts with Java	836	19	119	585		✓	✓		✓		✓	✓	✓	✓			✓	✓	🕮	🕮💻		some	all	✓	✓		✓
49	Java	The Java Way	650	24	213	377		✓	✓									✓				🕮💻			all		✓		✓
4	VB.NET	Intro. to Progr. Using VB.NET	Not available at time of publication																										
29	VB.NET	Microsoft VB.NET	581	13	54	327		✓	✓	✓	✓	✓	✓	✓	✓							💻			all		✓	✓	
45	VB.NET	Progr. in VB.NET	601	13	85	246		✓	✓		✓	✓	✓	✓								🕮⊙	⊙	some	some				
47	VB.NET	Learning Progr. Using VB.NET	591	11	0	335	✓	✓	✓	✓	✓		✓									⊙	⊙	some	some				
18	VB6	Progr. in VB 6.1	693	15	35	321	✓	✓	✓		✓		✓	✓	✓							🕮⊙	⊙	some	some				
31	VB6	VB6 from the Ground Up	931	26	307	0								✓	✓					✓		🕮							
34	VB6	Intro. to Progr. with VB 6.1	880	15	158	1714	✓	✓	✓		✓	✓	✓				✓	✓						some	some		✓		
38	VB6	The Visual Basic Coach	484	13	216	259	✓	✓					✓	✓	✓		✓	✓				🕮⊙💻		some	all		✓	✓	✓
46	VB6	Progr. with VB 6.1	935	13	0	815	✓	✓	✓		✓	✓	✓	✓				✓				💻	⊙						
15	Delphi	Progr. and PS with Delphi	650	17	69	324		✓	✓	✓	✓		✓	✓	✓	✓				✓									
9	Eiffel	OO Progr. in Eiffel	649	18	102	275	✓	✓	✓		✓										🕮	🕮		some	some				
13	Haskell	Haskell: The Craft of…	487	20	0	382	✓		✓		✓		✓								💻	💻	💻						
33	Haskell	An Intro. to Comp. with Haskell	Not available at time of publication																										
23	Matlab	MATLAB Progr. for Engineers	478	10	40	262		✓	✓		✓	✓		✓	✓						🕮	🕮💻		some	all				

Key to Symbols		
	🕮	Contained within text
	💻	Available via WWW
	⊙	On CD with text

Appendix 5. Curricular Content

#	Language	Brief Title	CS101I	CS111I	CS101O	CS111O	CS111F	CS101B	CS111A	CS111H
2		Simple Program Design	4/18	7/17	5/15	7/15	7/17	5/18	8/15	6/18
16		CS: An Overview	13/18	10/17	11/15	10/15	13/17	14/18	11/15	14/18
25		Using UML	1/18	2/17	0/15	1/15	1/17	0/18	2/15	1/18
1	C	Beginning With C	6/18	7/17	4/15	8/15	9/17	6/18	8/15	9/18
11	C	C Progr. Des. for Engineers	7/18	7/17	6/15	7/15	7/17	8/18	7/15	8/18
35	C	C How to Program	3/18	5/17	4/15	5/15	7/17	6/18	7/15	7/18
37	C	C Progr.: A Modern Approach	4/18	7/17	5/15	7/15	7/17	7/18	7/15	8/18
41	C	C by Dissection	3/18	5/17	5/15	5/15	5/17	5/18	6/15	5/18
44	C	Progr., PS & Abstr. with C	Not available at time of publication							
5	C++	C++ Program Design	6/18	10/17	6/15	8/15	6/17	5/18	8/15	8/18
6	C++	Struct. & OO PS Using C++	Not available at time of publication							
8	C++	C++ Programming…	6/18	8/17	6/15	8/15	9/17	8/18	9/15	9/18
10	C++	C++: An Intro. to Comp.	Not available at time of publication							
26	C++	A First Book of C++	6/18	7/17	7/15	6/15	7/17	7/18	7/15	6/18
27	C++	Computing Concepts with C++	9/18	11/17	9/15	12/15	12/17	10/18	12/15	11/18
32	C++	A CS Tapestry: ...with C++	4/18	6/17	7/15	6/15	7/17	7/18	8/15	6/18
48	C++	Problem Solving with C++	3/18	4/17	5/15	5/15	5/17	5/18	6/15	4/18
3	Java	Java Software Solutions	7/18	6/17	5/15	6/15	6/17	6/18	6/15	6/18
7	Java	Java: A Framework…	10/18	9/17	9/15	9/15	9/17	8/18	8/15	9/18
12	Java	Java: An Intro to CS & Progr.	Not available at time of publication							
14	Java	Objects First with Java	Not available at time of publication							
17	Java	PS with Java	6/18	6/17	6/15	6/15	7/17	6/18	6/15	6/18
19	Java	JAVA 2: Complete Ref.	3/18	4/17	4/15	4/15	4/17	3/18	3/15	3/18
20	Java	Java Progr: From Beginning	8/18	9/17	6/15	8/15	7/17	8/18	7/15	8/18
21	Java	Big Java	8/18	9/17	7/15	10/15	11/17	9/18	8/15	9/18
22	Java	Java Genesis	Not available at time of publication							
24	Java	Intro. to Java Progr. with Sun…	Not available at time of publication							
28	Java	Progr. and PS with Java	9/18	10/17	9/15	10/15	11/17	10/18	10/15	12/18
30	Java	The Object of Java	3/18	4/17	4/15	5/15	4/17	4/18	5/15	4/18
36	Java	Java How to Program	2/18	2/17	2/15	2/15	2/17	2/18	2/15	2/18
39	Java	Java, Java, Java	6/18	6/17	6/15	7/15	7/17	6/18	6/15	6/18
40	Java	An Invitation to CS - Java Ver.	16/18	14/17	13/15	12/15	13/17	13/18	13/15	15/18
42	Java	Intro. to OO Progr. with Java	6/18	8/17	7/15	8/15	8/17	7/18	8/15	8/18
43	Java	Computing Concepts with Java	7/18	8/17	7/15	9/15	8/17	7/18	8/15	8/18
49	Java	The Java Way	10/18	10/17	6/15	10/15	8/17	6/18	8/15	9/18
4	VB.NET	Intro. to Progr. Using VB.NET	Not available at time of publication							
29	VB.NET	Microsoft VB.NET	1/18	1/17	1/15	1/15	1/17	1/18	1/15	1/18
45	VB.NET	Progr. in VB.NET	2/18	3/17	3/15	3/15	2/17	2/18	2/15	2/18
47	VB.NET	Learning Progr. Using VB.NET	3/18	4/17	4/15	4/15	3/17	3/18	3/15	3/18
18	VB6	Progr. in VB 6.1	2/18	4/17	3/15	4/15	2/17	2/18	2/15	2/18
31	VB6	VB6 from the Ground Up	3/18	3/17	2/15	4/15	2/17	2/18	3/15	2/18
34	VB6	Intro. to Progr. with VB 6.1	2/18	3/17	3/15	3/15	2/17	2/18	3/15	2/18
38	VB6	The Visual Basic Coach	4/18	4/17	4/15	3/15	3/17	4/18	3/15	3/18
46	VB6	Progr. with VB 6.1	2/18	2/17	2/15	2/15	2/17	2/18	2/15	2/18
15	Delphi	Progr. and PS with Delphi	4/18	5/17	5/15	6/15	5/17	5/18	6/15	5/18
9	Eiffel	OO Progr. in Eiffel	4/18	7/17	5/15	6/15	6/17	5/18	6/15	6/18
13	Haskell	Haskell: The Craft of…	3/18	3/17	4/15	4/15	6/17	5/18	5/15	4/18
33	Haskell	An Intro. to Comp. with Haskell	Not available at time of publication							
23	Matlab	MATLAB Progr. for Engineers	6/18	6/17	4/15	6/15	5/17	4/18	4/15	5/18